UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/881,071	06/13/2001	Spencer M. Gold	SMQ-042	8019

959 7590 11/16/2004

LAHIVE & COCKFIELD, LLP.
28 STATE STREET
BOSTON, MA 02109

EXAMINER	
GERSTL, SHANE F	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 11/16/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/881,071	GOLD ET AL.	
	Examiner	Art Unit	
	Shane F Gerstl	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *19 August 2004*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-49* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-49* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *19 August 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-49 have been examined.

Papers Received

2. Receipt is acknowledged of amendment papers submitted, where the papers have been placed of record in the file.

3. The amendment has successfully overcome the objections to claim 22 and the drawings as well as the 35 USC 112 rejections.

4. The objections to the numbering of claims and the title remain as set forth below.

Specification

5. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed. The Examiner respectfully asserts that the title is not clearly indicative of the invention. The title does not give enough description as to what differentiates the invention from other systems and indicate what makes it a unique invention. The fact that the free list in the invention is capable of being flushed is not necessarily a new idea. Any free list is capable of being flushed, though in many systems this flushing may have adverse effects. The Examiner ask Applicant to please edit the title to be more descriptive of the invention in the sense of what is new in the art.

Claim Objections

6. Claims 5-17, 19-20, 34-45, and 47-48 are objected to because of the following informalities: they are in improper dependent form. MPEP § 608.01(n) states that a claim which depends from a dependent claim should not be separated by any claim

which does not also depend from said dependent claim. It is noted that any claim may depend on a previous independent claim. In general, applicant's sequence will not be changed.

Claim Rejections - 35 USC § 102

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

8. Claims 1-49 rejected under 35 U.S.C. 102(b) as being anticipated by Yeager (5,758,112).

9. In regard to claim 1, Yeager discloses in a microprocessor performing speculative instruction execution (figure 1 and column 2, lines 40-42), a method comprising the steps of:

a. providing a structure (figure 1, elements 204, 206, 208, and 210) to track register allocation for a first thread of said microprocessor; [Column 8, lines 27-33 show that a register mapping table maps or allocates physical registers to logical registers for integer instructions.]

b. and tracking a first set of pointers in said structure assigned to manage said register allocation for an instruction of said first thread of said microprocessor to prevent said register allocated as a destination operand for said instruction of said first thread from being overwritten before said instruction of said first thread retires, said first set of pointers includes at least two pointers

set apart by a fixed distance and move in unison up and down said structure. [Column 12, lines 52-56 show that a free list tracks available integer registers that are unused or free and may be assigned for an instruction destination. Column 13, lines 1-14 show a set of pointers (write pointer, read pointer, and graduation mask). Column 8 lines 1-17 show that a physical register is prevented from being overwritten since it written to only once before the value is sent back to a logical register and the physical register is free again. Column 14, lines 32-34 show that the physical registers are put back in the free list (and marked as free) when an instruction graduates or retires. Thus the physical register is not overwritten until the instruction retires. Column 13, lines 1-14 show a set of pointers (write pointer, read pointer, and graduation mask or retire pointer). Column 14, lines 54-55 show that the write pointer is incremented by the number of instructions that graduate (retire) each cycle. Column 15, lines 1-2 show that the read pointer is incremented by the number of free registers assigned each cycle. In the case of a cycle where the same number of instructions retire as the number of free registers that are assigned, the read and write pointers move in unison up and down said structure (up when incrementing and down when both pointers wrap around the circular FIFO (column 12, lines 58-60)). Further, at any given point there is a fixed distance between the read and write pointers. While there are times where the read and write pointers do not move in unison, there clearly are times when the pointers do move in unison. Applicant has not claimed where the pointers move in unison at all times.]

10. In regard to claim 2, Yeager discloses the method of claim 1, further comprising the step of tracking a second set of pointers in said structure assigned to manage a register allocation for an instruction of a second thread of said microprocessor to prevent a register allocated as a destination operand for said instruction of said second thread from being overwritten before said instruction of said second thread retires, whereby said first set of pointers and said second set of pointers track independently of each other. [The sections cited above also show that floating-point instructions have renamed registers and a separate free list (figure 1, element 208) and mapping table (element 204) for the same purpose as integer instructions. Figure 1 shows that the mapping and list structures for each are in two separate flows of control and thus two separate threads.]

11. In regard to claim 3, Yeager discloses the method of claim 1, wherein said structure comprises pointers to said registers allocated and pointers to registers deallocated, where said registers allocated and said registers deallocated are physical registers that operate as a destination operand for said instructions executing on a multithreading microprocessor where said destination operands identifies where data resulting from logical operations are to be written. [Column 12, lines 24-56 show that the registers are pointed to in either a free list, which tracks deallocated registers or registers that are unused and may be assigned, or a mapping table, which points to allocated registers as described above. As shown above the registers pointed to are physical destination registers. Figure 1 shows that there are multiple threads with a flow of control for integer instructions and a flow of control for floating point instructions.]

12. In regard to claim 4, Yeager discloses the method of claim 1, wherein said first set of pointers comprises a read pointer, a write pointer, and a retire pointer where said read and write pointers are set apart by a fixed distance and move in unison up and down said structure. [Column 13, lines 1-14 show a set of pointers (write pointer, read pointer, and graduation mask or retire pointer). Column 14, lines 54-55 show that the write pointer is incremented by the number of instructions that graduate (retire) each cycle. Column 15, lines 1-2 show that the read pointer is incremented by the number of free registers assigned each cycle. In the case of a cycle where the same number of instructions retire as the number of free registers that are assigned, the read and write pointers move in unison up and down said structure (up when incrementing and down when both pointers wrap around the circular FIFO (column 12, lines 58-60)). At any given point there is a fixed distance between the read and write pointers. Column 14, lines 39-43 and column 13, lines 12-14 show that the graduation mask indicates or points to the graduating (retiring) instruction that is releasing a register and is thus suitably called a retire pointer.]

13. In regard to claim 5, Yeager discloses the method of claim 2, wherein said second set of pointers comprises a read pointer, a write now pointer, and a retire pointer where said read and write pointers are set apart by a fixed distance and move in unison up and down said structure. [Column 13, lines 1-14 show a set of pointers (write pointer, read pointer, and graduation mask or retire pointer). Column 14, lines 20-24 show that the write pointer gives the next entry to be written, or to be written now and is then a write now pointer. Column 14, lines 54-55 show that the write pointer is

incremented by the number of instructions that graduate (retire) each cycle. Column 15, lines 1-2 show that the read pointer is incremented by the number of free registers assigned each cycle. In the case of a cycle where the same number of instructions retire as the number of free registers that are assigned, the read and write pointers move in unison up and down said structure (up when incrementing and down when both pointers wrap around the circular FIFO (column 12, lines 58-60)). At any given point there is a fixed distance between the read and write pointers. Column 14, lines 39-43 and column 13, lines 12-14 show that the graduation mask indicates or points to the graduating (retiring) instruction that is releasing a register and is thus suitably called a retire pointer. As shown above, the integer and floating-point threads are both renamed in a similar manner and thus the above sections are pertinent to both threads.]

14. In regard to claim 6, Yeager discloses the method of claim 4, wherein said read pointer of said first set of pointers indicates said physical register location awaiting said register allocation as a said destination operand to identify where data should be written when said instruction of said first thread is executed by said microprocessor. [Column 14, lines 56-59 show that the read pointer indicates a destination register (shown above to be physical) for allocation. This destination register is then written to on execution of the instruction in the thread since it is the instruction result destination.]

15. In regard to claim 7, Yeager discloses the method of claim 4, wherein said write pointer of said first set of pointers indicates said physical register location of said register allocated as a said destination operand to identify where data should be written for said instruction of said thread that committed. [As shown above, the write pointer

points to an entry to next write to. Since the result of an instruction is ready at the commit stage of a superscalar processor, this is when the destination is pointed to for writing.]

16. In regard to claim 8, Yeager discloses the method of claim 4, wherein said retire pointer of said first set of pointers indicates said physical register location of said register allocated as a said destination operand for said instruction of said thread that is next to be retired. [As shown above, the retire pointer (graduation mask) identifies the instruction that is releasing a physical destination register (and thus indicates the physical register) since the retiring instruction is writing the data of that physical register into the appropriate logical register.]

17. In regard to claim 9, Yeager discloses the method of claim 5, wherein said read pointer of said first set of pointers indicates said physical register location awaiting said register allocation as a said destination operand to identify where data should be written when said instruction of said second thread is executed by said microprocessor. [Column 14, lines 56-59 show that the read pointer indicates a destination register (shown above to be physical) for allocation. This destination register is then written to on execution of the instruction in the thread since it is the instruction result destination. As shown above, the integer and floating-point threads are both renamed in a similar manner and thus the above sections are pertinent to both threads.]

18. In regard to claim 10, Yeager discloses the method of claim 5, wherein said write pointer of said first set of pointers indicates said physical register location of said register allocated as a said destination operand to identify where data should be written

for said instruction of said second thread that committed. [As shown above, the write pointer points to an entry to next write to. Since the result of an instruction is ready at the commit stage of a superscalar processor, this is when the destination is pointed to for writing.]

19. In regard to claim 11, Yeager discloses the method of claim 5, wherein said retire pointer of said first set of pointers indicates said physical register location of said register allocated as a said destination operand for said instruction of said second thread that is next to be retired. [As shown above, the retire pointer (graduation mask) identifies the instruction that is releasing a physical destination register (and thus indicates the physical register) since the retiring instruction is writing the data of that physical register into the appropriate logical register. As shown above, the integer and floating-point threads are both renamed in a similar manner and thus the above sections are pertinent to both threads.]

20. In regard to claim 12, Yeager discloses the method of claim 4, wherein the number of physical register pointers between said read pointer and said retire pointer of said first set of pointers indicates said physical registers available for said register allocation for said first thread of said microprocessor. [Column 12, lines 57-60 show that the free list is incorporated as a circular buffer. As shown previously, the read pointer points to the next register to be allocated a physical register, the write pointer points to the register to write to after a commit, and the retire pointer points to an instruction to next retire and thus indirectly points to a physical register to free. This means that the circular buffer is configured where the read pointer is ahead of the write

pointer, which is ahead of the retire pointer. Since the buffer is circular, the entries of the buffer ahead of the read pointer all the way up and wrapped around to the retire pointer, are available and not allocated.]

21. In regard to claim 13, Yeager discloses the method of claim 5, wherein the number of physical register pointers between said read pointer and said retire pointer of said second set of pointers indicates said physical registers available for said register allocation for said second thread of said microprocessor. [Column 12, lines 57-60 show that the free list is incorporated as a circular buffer. As shown previously, the read pointer points to the next register to be allocated a physical register, the write pointer points to the register to write to after a commit, and the retire pointer points to an instruction to next retire and thus indirectly points to a physical register to free. This means that the circular buffer is configured where the read pointer is ahead of the write pointer, which is ahead of the retire pointer. Since the buffer is circular, the entries of the buffer ahead of the read pointer all the way up and wrapped around to the retire pointer, are available and not allocated.]

22. In regard to claim 14, Yeager discloses the method of claim 4, wherein the number of physical register pointers between said retire pointer and said write pointer of said first set of pointers indicates said registers allocated to said destination operand for a plurality of instructions of said first thread that are to become available for reallocation upon retirement of said plurality of instructions. [Since as shown above, the retirement pointer points to instructions with physical registers that are next to be freed or available for allocation. Since the write pointer points to destinations to be written before

retirement, the number of pointers between these write and retire pointers are inherently indications of registers allocated to destination operands that will become available upon retirement.]

23. In regard to claim 15, Yeager discloses the method of claim 5, wherein the number of physical register pointers between said retire pointer and said write pointer of said second set of pointers indicates said physical registers allocated to said destination operand for a plurality of instructions of said second thread that are to become available for reallocation upon retirement of said plurality of instructions. [Since as shown above, the retirement pointer points to instructions with physical registers that are next to be freed or available for allocation. Since the write pointer points to destinations to be written before retirement, the number of pointers between these write and retire pointers are inherently indications of registers allocated to destination operands that will become available upon retirement.]

24. In regard to claim 16, Yeager discloses the method of claim 4, wherein said registers allocated for said plurality of instructions of said first thread of said microprocessor that have not yet committed is defined by the number of physical register pointers bounded by said read pointer and said retire pointer minus said fixed distance between said read pointer and said write pointer of said first set of pointers. [Since the read pointer leads the write pointer, which leads the retire pointer, as shown above, the pointers between the read and retire pointers minus the pointers between the read and write pointers (which is a fixed distance at any one time) inherently leaves

the pointers between the write and retire pointers which have been shown to be the registers allocated and not yet committed above.]

25. In regard to claim 17, Yeager discloses the method of claim 5, wherein said register allocated for said plurality of instructions said second thread of said microprocessor that have not yet committed is defined by the number of physical register pointers bounded by said read pointer and said retire pointer minus said fixed distance between said read pointer and said write pointer of said second set of pointers. [Since the read pointer leads the write pointer, which leads the retire pointer, as shown above, the pointers between the read and retire pointers minus the pointers between the read and write pointers (which is a fixed distance at any one time) inherently leaves the pointers between the write and retire pointers which have been shown above to be the registers allocated and not yet committed.]

26. In regard to claim 18, Yeager discloses the method of claim 1, wherein said method of register allocation is performed in a modulo-8 memory array. [Column 12, lines 57-61 show that the free list is embodied on an eight-entry circular buffer, which is the conventional definition in the art for a modulo-8 memory where a maximum of eight elements can be in flight at one time.]

27. In regard to claim 19, Yeager discloses the method of claim 16, further comprising the step of restoring said register allocated for said instruction of said first thread of said microprocessor that has not yet committed to its previous state in said first thread of said microprocessor by pointing said read pointer of said first set of pointers to said physical register pointer allocated to said physical register location

corresponding to said instruction being flushed by said microprocessor. [Column 16, lines 24-42 show that in case of an exception the mappings of register allocation must be restored and this is done by aborting or flushing subsequent instructions (which have not yet committed) and adjusting the read pointer.]

28. In regard to claim 20, Yeager discloses the method of claim 17, further comprising the step of restoring said register allocated for said instruction of said second thread of said microprocessor that has not yet committed to its previous state in said second thread of said microprocessor by pointing said read pointer of said second set of pointers to said physical register pointer allocated to said physical register location corresponding to said instruction being flushed by said microprocessor. [Column 16, lines 24-42 show that in case of an exception the mappings of register allocation must be restored and this is done by aborting or flushing subsequent instructions (which have not yet committed) and adjusting the read pointer. As shown above and throughout the disclosure, the renaming techniques are similar for both threads.]

29. In regard to claim 21, Yeager discloses in a multithreading microprocessor performing speculative instruction execution, a method comprising the steps of:

a. providing a structure (figure 1, elements 204, 206, 208, and 210) to track register allocation for a first thread and a second thread of said multithreading microprocessor; [Column 8, lines 27-33 show that a register mapping table maps or allocates physical registers to logical registers for integer instructions.]

b. tracking a first set of pointers in said structure assigned to manage said register allocation for an instruction of said first thread of said multithreading processor to prevent a register allocated as a destination operand for said instruction of said first thread from being overwritten before said instruction of said first thread retires; [Column 12, lines 52-56 show that a free list tracks available integer registers that are unused or free and may be assigned for an instruction destination. Column 13, lines 1-14 show a set of pointers (write pointer, read pointer, and graduation mask). Column 8 lines 1-17 show that a physical register is prevented from being overwritten since it written to only once before the value is sent back to a logical register and the physical register is free again. Column 14, lines 32-34 show that the physical registers are put back in the free list (and marked as free) when an instruction graduates or retires. Thus the physical register is not overwritten until the instruction retires.]

c. and tracking a second set of pointers in said structure assigned to manage said register allocation for an instruction of said second thread of said multithreading processor to prevent a register allocated as a destination operand for said instruction of said second thread from being overwritten before said instruction of said second thread retires, whereby said first set of pointers and said second set of pointers track independently of each other. [The sections cited above also show that floating-point instructions have renamed registers and a separate free list (figure 1, element 208) and mapping table (element 204) for the same purpose as integer instructions (not separate structures but separate

parts). Figure 1 shows that the mapping and list structures for each are in two separate flows of control and thus two separate threads.]

30. In regard to claim 22, Yeager discloses a semiconductor device (figure 1, microprocessor) having a plurality of physical registers that are assigned as destination registers for instructions to be executed by a microprocessor performing out-of-order execution (column 1, lines 56-59), comprising:

a. a first module (figure 1, elements 204, 206, 208, and 210) providing a structure for holding information identifying available physical registers for said microprocessor; [Column 8, lines 27-33 show that a register mapping table maps or allocates physical registers to logical registers for integer instructions.]

b. a first set of register pointers assigned to a first portion of said structure to track said physical registers assigned as said destination registers for a first thread of said microprocessor, said first set of register pointers includes a retire row pointer to identify where a pointer pointing to at least one of said plurality of physical registers assigned as a destination register for an instruction in said first thread that is next to be retired and a read pointer to identify where a pointer pointing to an available physical register available for assignment as a destination operand for an instruction for said first thread, wherein when said microprocessor issues a flush request for said instruction in said first thread, moving said read pointer of said first set of register pointers to said physical register assigned as said destination register for said instruction being flushed in said first thread to restore said physical register to a previous state. [Column 12,

lines 52-56 show that a free list tracks available integer registers that are unused or free and may be assigned for an instruction destination. Column 13, lines 1-14 show a set of pointers (write pointer, read pointer, and graduation mask). Column 8 lines 1-17 show that a physical register is prevented from being overwritten since it written to only once before the value is sent back to a logical register and the physical register is free again. Column 14, lines 32-34 show that the physical registers are put back in the free list (and marked as free) when an instruction graduates or retires. Thus the physical register is not overwritten until the instruction retires. Column 16, lines 24-42 show that in case of an exception the mappings of register allocation must be restored and this is done by aborting or flushing subsequent instructions (which have not yet committed) and adjusting the read pointer. Column 13, lines 12-14 and column 14, lines 39-43 show that a graduation mask or retire pointer indicate the next graduating or retiring instruction from a group, where the one selected can be viewed as of a certain row and thus a row pointer is used. The included dictionary definition of "pointer" (third definition) defines a pointer to be a physical or symbolic identifier of a unique target. With this definition, the graduation mask of Yeager, which is shown in column 13, lines 12-13 to identify which instructions graduate or retire, is in fact a retire pointer. The graduation mask identifies a unique target, the instructions to graduate or retire.]

31. In regard to claim 23, Yeager discloses the semiconductor device of claim 22, further comprising a second set of register pointers assigned to a second portion of said

structure to track said physical registers assigned as said destination registers for a second thread of said microprocessor and when said microprocessor issues a flush request for an instruction in said second thread, moving a read pointer of said second set of register pointers to said physical register assigned as said destination register for said instruction being flushed in said second thread to restore said physical register to a previous state. [The sections cited above also show that floating-point instructions have renamed registers and a separate free list (figure 1, element 208) and mapping table (element 204) for the same purpose as integer instructions. Figure 1 shows that the mapping and list structures for each are in two separate flows of control and thus two separate threads.]

32. In regard to claim 24, Yeager discloses the semiconductor device of claim 22, wherein said structure comprises a free physical register list for said identification of said available physical requests for said microprocessor (as shown above).

33. In regard to claim 25, Yeager discloses the semiconductor device of claim 23, wherein said first set of register pointers move independently of said second set of register pointers, wherein said first set of register pointers identify said physical registers assigned to instructions in said first thread of said microprocessor that have not been committed and said second set of register pointers identify said physical registers assigned to instructions in said second thread of said microprocessor that have not been committed. [As shown above the two threads are independent since they have two flows of control. The set of pointers for each thread comprise a read pointer (column 13, lines 1-6. This pointer identifies registers of instructions for allocation

(column 14, lines 56-59) that have not yet been committed, since the committing does not occur until the write pointer identifies a register for writing.]

34. In regard to claim 26, Yeager discloses the semiconductor device of claim 22, wherein said first set of register pointers further comprises, a write row pointer, wherein said write row pointer identifies where a pointer pointing to said physical register of an instruction in said first thread should be written when said instruction commits. [Column 13, lines 14-16 show that a write row pointer is used. Column 14, lines 20-24 show that this is the place to next write, which happens on a commit when the result is ready.]

35. In regard to claim 27, Yeager discloses the semiconductor device of claim 23, wherein said second set of register pointers further comprises, a write row pointer and a retire row pointer, wherein said write row pointer identifies where a pointer pointing to said physical register of an instruction in said second thread that is committed should be written, and said retire row pointer identifies where a pointer pointing to said physical register of an instruction in said second thread that is next to be retired. [Column 13, lines 14-16 show that a write row pointer is used. Column 14, lines 20-24 show that this is the place to next write, which happens on a commit when the result is ready. Column 13, lines 12-14 and column 14, lines 39-43 show that a graduation mask or retire pointer indicate the next graduating or retiring instruction from a group, where the one selected can be viewed as of a certain row and thus a row pointer is used. As shown above the renaming works similar for both threads.]

36. In regard to claim 28, Yeager discloses a semiconductor device having a plurality of physical registers that are assigned as destination registers for instructions to be executed by a microprocessor performing out-of-order execution, comprising:

a. a first module (figure 1, elements 204, 206, 208, and 210) providing a structure for holding information identifying available physical registers for said microprocessor; Column 8, lines 27-33 show that a register mapping table maps or allocates physical registers to logical registers for integer instructions.

b. a first set of register pointers assigned to a first portion of said structure to track said physical registers assigned as said destination registers for a first thread of said microprocessor and when said microprocessor issues a flush request for an instruction in said first thread, moving a read pointer of said first set of register pointers to said physical register assigned as said destination register for said instruction being flushed in said first thread to restore said physical register to a previous state; [Column 12, lines 52-56 show that a free list tracks available integer registers that are unused or free and may be assigned for an instruction destination. Column 13, lines 1-14 show a set of pointers (write pointer, read pointer, and graduation mask). Column 8 lines 1-17 show that a physical register is prevented from being overwritten since it written to only once before the value is sent back to a logical register and the physical register is free again. Column 14, lines 32-34 show that the physical registers are put back in the free list (and marked as free) when an instruction graduates or retires. Thus the physical register is not overwritten until the instruction retires. Column 16,

lines 24-42 show that in case of an exception the mappings of register allocation must be restored and this is done by aborting or flushing subsequent instructions (which have not yet committed) and adjusting the read pointer.]

d. and a second set of register pointers assigned to a second portion of said structure to track said physical registers assigned as said destination registers for a second thread of said microprocessor and when said microprocessor issues a flush request for an instruction in said second thread, moving a read pointer of said second set of register pointers to said physical register assigned as said destination register for said instruction being flushed in said second thread to restore said physical register to a previous state. [The sections cited above also show that floating-point instructions have renamed registers and a separate free list (figure 1, element 208) and mapping table (element 204) for the same purpose as integer instructions (separate meaning different parts of the same structure). Figure 1 shows that the mapping and list structures for each are in two separate flows of control and thus two separate threads.]

37. In regard to claim 29, Yeager discloses a computer readable medium (figure 1, element 102) holding computer executable instructions for performing a method in a microprocessor performing speculative instruction execution (figure 1 and column 2, lines 40-42), said method comprising the steps of:

a. providing a structure (figure 1, elements 204, 206, 208, and 210) to track register allocation for a first thread of said microprocessor; [Column 8, lines 27-

33 show that a register mapping table maps or allocates physical registers to logical registers for integer instructions.]

b. and tracking a first set of pointers in said structure assigned to manage said register allocation for an instruction of said first thread of said microprocessor to prevent a register allocated as a destination operand for said instruction of said first thread from being overwritten before said instruction of said first thread retires, said first set of pointers includes at least two pointers set apart by a fixed distance and move in unison up and down said structure. [Column 12, lines 52-56 show that a free list tracks available integer registers that are unused or free and may be assigned for an instruction destination. Column 13, lines 1-14 show a set of pointers (write pointer, read pointer, and graduation mask). Column 8 lines 1-17 show that a physical register is prevented from being overwritten since it written to only once before the value is sent back to a logical register and the physical register is free again. Column 14, lines 32-34 show that the physical registers are put back in the free list (and marked as free) when an instruction graduates or retires. Thus the physical register is not overwritten until the instruction retires. Column 14, lines 54-55 show that the write pointer is incremented by the number of instructions that graduate (retire) each cycle. Column 15, lines 1-2 show that the read pointer is incremented by the number of free registers assigned each cycle. In the case of a cycle where the same number of instructions retire as the number of free registers that are assigned, the read and write pointers move in unison up and

down said structure (up when incrementing and down when both pointers wrap around the circular FIFO (column 12, lines 58-60)). Further, at any given point there is a fixed distance between the read and write pointers. While there are times where the read and write pointers do not move in unison, there clearly are times when the pointers do move in unison. Applicant has not claimed where the pointers move in unison at all times.]

38. In regard to claim 30, Yeager discloses the computer readable medium of claim 29 further comprising the step of tracking a second set of pointers in said structure assigned to manage said register allocation for an instruction of a second thread of said microprocessor to prevent a register allocated as a destination operand for said instruction of said second thread from being overwritten before said instruction of said second thread retires, whereby said first set of pointers and said second set of pointers track independently of each other. [The sections cited above also show that floating-point instructions have renamed registers and a separate free list (figure 1, element 208) and mapping table (element 204) for the same purpose as integer instructions. Figure 1 shows that the mapping and list structures for each are in two separate flows of control and thus two separate threads.]

39. In regard to claim 31, Yeager discloses the computer readable medium of claim 29, wherein said structure comprises pointers to said registers allocated and pointers to registers deallocated, where said registers allocated and said registers deallocated are physical registers that operate as a destination operands for said instruction of said first thread executing on said microprocessor where said destination operands identifies

where data resulting from logical operations are to be written. [Column 12, lines 24-56 show that the registers are pointed to in either a free list, which tracks deallocated registers or registers that are unused and may be assigned, or a mapping table, which points to allocated registers as described above. As shown above the registers pointed to are physical destination registers. Figure 1 shows that there are multiple threads with a flow of control for integer instructions and a flow of control for floating point instructions.]

40. In regard to claim 32, Yeager discloses the computer readable medium of claim 29, wherein said first set of pointers comprises a read pointer, a write pointer, and a retire pointer wherein said read pointer and said write pointer are set apart by a fixed distance and move in unison up and down said structure. [Column 13, lines 1-14 show a set of pointers (write pointer, read pointer, and graduation mask or retire pointer). Column 14, lines 54-55 show that the write pointer is incremented by the number of instructions that graduate (retire) each cycle. Column 15, lines 1-2 show that the read pointer is incremented by the number of free registers assigned each cycle. In the case of a cycle where the same number of instructions retire as the number of free registers that are assigned, the read and write pointers move in unison up and down said structure (up when incrementing and down when both pointers wrap around the circular FIFO (column 12, lines 58-60)). At any given point there is a fixed distance between the read and write pointers. Column 14, lines 39-43 and column 13, lines 12-14 show that the graduation mask indicates or points to the graduating (retiring) instruction that is releasing a register and is thus suitably called a retire pointer.]

41. In regard to claim 33, Yeager discloses the computer readable medium of claim 29, wherein said second set of pointers comprises a read pointer, a write pointer, and a retire pointer wherein said read pointer and said write pointer are set apart by a fixed distance and move in unison up and down said structure. [Column 13, lines 1-14 show a set of pointers (write pointer, read pointer, and graduation mask or retire pointer). Column 14, lines 20-24 show that the write pointer gives the next entry to be written, or to be written now and is then a write now pointer. Column 14, lines 54-55 show that the write pointer is incremented by the number of instructions that graduate (retire) each cycle. Column 15, lines 1-2 show that the read pointer is incremented by the number of free registers assigned each cycle. In the case of a cycle where the same number of instructions retire as the number of free registers that are assigned, the read and write pointers move in unison up and down said structure (up when incrementing and down when both pointers wrap around the circular FIFO (column 12, lines 58-60)). At any given point there is a fixed distance between the read and write pointers. Column 14, lines 39-43 and column 13, lines 12-14 show that the graduation mask indicates or points to the graduating (retiring) instruction that is releasing a register and is thus suitably called a retire pointer. As shown above, the integer and floating-point threads are both renamed in a similar manner and thus the above sections are pertinent to both threads.]

42. In regard to claim 34, Yeager discloses the computer readable medium of claim 32, wherein said read pointer of said first set of pointers indicates said physical register location awaiting said register allocation as a said destination operand to identify where

data should be written when said instruction of said first thread is executed by said microprocessor. [Column 14, lines 56-59 show that the read pointer indicates a destination register (shown above to be physical) for allocation. This destination register is then written to on execution of the instruction in the thread since it is the instruction result destination.]

43. In regard to claim 35, Yeager discloses the computer readable medium of claim 32, wherein said write pointer of said first set of pointers indicates said physical register location of said register allocated as a said destination operand to identify where data should be written for said instruction of said first thread that committed. [As shown above, the write pointer points to an entry to next write to. Since the result of an instruction is ready at the commit stage of a superscalar processor, this is when the destination is pointed to for writing.]

44. In regard to claim 36, Yeager discloses the computer readable medium of claim 32, wherein said retire pointer of said first set of pointers indicates said physical register location of said register allocated as a said destination operand for said instruction of said first thread that is next to be retired. [As shown above, the retire pointer (graduation mask) identifies the instruction that is releasing a physical destination register (and thus indicates the physical register) since the retiring instruction is writing the data of that physical register into the appropriate logical register.]

45. In regard to claim 37, Yeager discloses the computer readable medium of claim 33, wherein said read pointer indicates said physical register location awaiting said register allocation as a said destination operand to identify where data should be written

when said instruction of said second thread is executed by said microprocessor. [Column 14, lines 56-59 show that the read pointer indicates a destination register (shown above to be physical) for allocation. This destination register is then written to on execution of the instruction in the thread since it is the instruction result destination. As shown above, the integer and floating-point threads are both renamed in a similar manner and thus the above sections are pertinent to both threads.]

46. In regard to claim 38, Yeager discloses the computer readable medium of claim 33, wherein said write pointer of said first set of pointers indicates said physical register location of said register allocated as a said destination operand to identify where data should be written for said instruction of said second thread that committed. [As shown above, the write pointer points to an entry to next write to. Since the result of an instruction is ready at the commit stage of a superscalar processor, this is when the destination is pointed to for writing.]

47. In regard to claim 39, Yeager discloses the computer readable medium of claim 33, wherein said retire pointer of said first set of pointers indicates said physical register location of said register allocated as a said destination operand for said instruction of said second thread that is next to be retired. [As shown above, the retire pointer (graduation mask) identifies the instruction that is releasing a physical destination register (and thus indicates the physical register) since the retiring instruction is writing the data of that physical register into the appropriate logical register. As shown above, the integer and floating-point threads are both renamed in a similar manner and thus the above sections are pertinent to both threads.]

48. In regard to claim 40, Yeager discloses the computer readable medium of claim 32, wherein the number of physical register pointers between said read pointer and said retire pointer of said first set of pointers indicates said physical registers available for said register allocation for said first thread of said microprocessor. [Column 12, lines 57-60 show that the free list is incorporated as a circular buffer. As shown previously, the read pointer points to the next register to be allocated a physical register, the write pointer points to the register to write to after a commit, and the retire pointer points to an instruction to next retire and thus indirectly points to a physical register to free. This means that the circular buffer is configured where the read pointer is ahead of the write pointer, which is ahead of the retire pointer. Since the buffer is circular, the entries of the buffer ahead of the read pointer all the way up and wrapped around to the retire pointer, are available and not allocated.]

49. In regard to claim 41, Yeager discloses the computer readable medium of claim 33, wherein the number of physical register pointers between said read pointer and said retire pointer of said second set of pointers indicates said physical registers available for said register allocation for said second thread of said microprocessor. [Column 12, lines 57-60 show that the free list is incorporated as a circular buffer. As shown previously, the read pointer points to the next register to be allocated a physical register, the write pointer points to the register to write to after a commit, and the retire pointer points to an instruction to next retire and thus indirectly points to a physical register to free. This means that the circular buffer is configured where the read pointer is ahead of the write pointer, which is ahead of the retire pointer. Since the buffer is circular, the entries of

the buffer ahead of the read pointer all the way up and wrapped around to the retire pointer, are available and not allocated.]

50. In regard to claim 42, Yeager discloses the computer readable medium of claim 32, wherein the number of physical register pointers between said retire pointer and said write pointer of said first set of pointers indicates said registers allocated to said destination operand for a plurality of instructions of said first thread that are to become available for reallocation upon retirement of said plurality of instructions. [Since as shown above, the retirement pointer points to instructions with physical registers that are next to be freed or available for allocation. Since the write pointer points to destinations to be written before retirement, the number of pointers between these write and retire pointers are inherently indications of registers allocated to destination operands that will become available upon retirement.]

51. In regard to claim 43, Yeager discloses the computer readable medium of claim 33, wherein the number of physical register pointers between said retire pointer and said write pointer of said second set of pointers indicates said physical registers allocated to said destination operand for a plurality of instructions of said second thread that are to become available for reallocation upon retirement of said plurality of instructions. [Since as shown above, the retirement pointer points to instructions with physical registers that are next to be freed or available for allocation. Since the write pointer points to destinations to be written before retirement, the number of pointers between these write and retire pointers are inherently indications of registers allocated to destination operands that will become available upon retirement.]

52. In regard to claim 44, Yeager discloses the computer readable medium of claim 32, wherein said register allocated for said instructions of said first thread of said microprocessor that have not yet committed is defined by the number of physical register pointers bounded by said read pointer and said retire pointer minus said fixed distance between said read pointer and said write pointer of said first set of pointers. [Since the read pointer leads the write pointer, which leads the retire pointer, as shown above, the pointers between the read and retire pointers minus the pointers between the read and write pointers (which is a fixed distance at any one time) inherently leaves the pointers between the write and retire pointers which have been shown to be the registers allocated and not yet committed above.]

53. In regard to claim 45, Yeager discloses the computer readable medium of claim 33, wherein said register allocated for said instructions said second thread of said microprocessor that have not yet committed is defined by the number of physical register pointers bounded by said read pointer and said retire pointer minus said fixed distance between said read pointer and said write pointer of said second set of pointers. [Since the read pointer leads the write pointer, which leads the retire pointer, as shown above, the pointers between the read and retire pointers minus the pointers between the read and write pointers (which is a fixed distance at any one time) inherently leaves the pointers between the write and retire pointers which have been shown above to be the registers allocated and not yet committed.]

54. In regard to claim 46, Yeager discloses the computer readable medium of claim 29, wherein said method of register allocation is performed in a modulo-8 memory

array. [Column 12, lines 57-61 show that the free list is embodied on an eight-entry circular buffer, which is the conventional definition in the art for a modulo-8 memory where a maximum of eight elements can be in flight at one time.]

55. In regard to claim 47, Yeager discloses the computer readable medium of claim 44, further comprising the step of restoring said register allocated for said instruction of said first thread of said microprocessor that has not yet committed to its previous state in said first thread of said microprocessor by pointing said read pointer of said first set of pointers to said physical register pointer allocated to said physical register location corresponding to said instruction being flushed by said microprocessor. [Column 16, lines 24-42 show that in case of an exception the mappings of register allocation must be restored and this is done by aborting or flushing subsequent instructions (which have not yet committed) and adjusting the read pointer.]

56. In regard to claim 48, Yeager discloses the computer readable medium of claim 45, further comprising the step of restoring said register allocated for said instruction of said second thread of said microprocessor that has not yet committed to its previous state in said second thread of said microprocessor by pointing said read pointer of said second set of pointers to said physical register pointer allocated to said physical register location corresponding to said instruction being flushed by said microprocessor. [Column 16, lines 24-42 show that in case of an exception the mappings of register allocation must be restored and this is done by aborting or flushing subsequent instructions (which have not yet committed) and adjusting the read pointer. As shown

above and throughout the disclosure, the renaming techniques are similar for both threads.]

57. In regard to claim 49, Yeager discloses a computer readable medium (figure 1, element 102) holding computer executable instructions for performing a method in a multithreading microprocessor performing speculative instruction execution (figure 1 and column 2, lines 40-42), said method comprising the steps of:

a. providing a structure (figure 1, elements 204, 206, 208, and 210) to track register allocation for a first thread and a second thread of said multithreading microprocessor; [Column 8, lines 27-33 show that a register mapping table maps or allocates physical registers to logical registers for integer instructions.]

b. tracking a first set of pointers in said structure assigned to manage said register allocation for an instruction of said first thread of said multithreading processor to prevent a register allocated as a destination operand for said instruction of said first thread from being overwritten before said instruction of said first thread retires; [Column 12, lines 52-56 show that a free list tracks available integer registers that are unused or free and may be assigned for an instruction destination. Column 13, lines 1-14 show a set of pointers (write pointer, read pointer, and graduation mask). Column 8 lines 1-17 show that a physical register is prevented from being overwritten since it written to only once before the value is sent back to a logical register and the physical register is free again. Column 14, lines 32-34 show that the physical registers are put back in

the free list (and marked as free) when an instruction graduates or retires. Thus the physical register is not overwritten until the instruction retires.]

c. and tracking a second set of pointers in said structure assigned to manage said register allocation for an instruction of said second thread of said multithreading processor to prevent a register allocated as a destination operand for said instruction of said second thread from being overwritten before said instruction of said second thread retires, whereby said first set of pointers and said second set of pointers track independently of each other. [The sections cited above also show that floating-point instructions have renamed registers and a separate (same structure different parts) free list (figure 1, element 208) and mapping table (element 204) for the same purpose as integer instructions. Figure 1 shows that the mapping and list structures for each are in two separate flows of control and thus two separate threads.]

Response to Arguments

58. Applicant's arguments filed 8/19/04 have been fully considered but they are not persuasive.

59. Applicant argues that the Yeager patent moves a write pointer in increments determined by the number of instructions which graduate during each clock cycle and moves the read pointer in increments by the number of free registers assigned during each clock cycle and thus does not anticipate claim 1. The Examiner recognizes what the Applicant asserts to be true, however this does not mean that the Yeager reference does not anticipate claim 1 as will be described further below. Applicant also argues

that Yeager patent does not disclose two pointers set apart by a fixed distance that move in unison up and down a structure to track register allocation of a first thread of a microprocessor, but instead discloses two pointers that move independently. The Examiner points out with clarifications the argument made in the previous Office Action regarding the former claim 4.

60. Column 13, lines 1-14 show a set of pointers (write pointer, read pointer, and graduation mask or retire pointer). Column 14, lines 54-55 show that the write pointer is incremented by the number of instructions that graduate (retire) each cycle. Column 15, lines 1-2 show that the read pointer is incremented by the number of free registers assigned each cycle. [All as previously described] In the case of a cycle where the same number of instructions retire as the number of free registers that are assigned, the read and write pointers move in unison up and down said structure (up when incrementing and down when both pointers wrap around the circular FIFO (column 12, lines 58-60)). Further, at any given point there is a fixed distance between the read and write pointers. While there are times where the read and write pointers do not move in unison, there clearly are times when the pointers do move in unison. Applicant has not claimed where the pointers move in unison at all times.

61. Applicant argues that Yeager does not anticipate claim 21 because Yeager discloses two independent free register lists, each with a set of pointers, and asserts that claim 21 recites a single structure with multiple sets of pointers where the single structure handles both threads. The Examiner asserts that claim 21 does not exclusively claim that which the Applicant argues. For instance, the first indent of claim

21 states "providing a structure to track register allocation for a first thread and a second thread of said multithreading microprocessor". Now, this can be interpreted as the Applicant describes, however the Examiner has interpreted the claim to mean providing a structure, one each for a first thread and a second thread, of which the argument presented for claim 21 meets. Indeed with this interpretation there is a structure for the first thread and a structure for the second thread or for the first and second thread. Applicant also points out that the Examiner can validly view elements 204, 206, 208, and 210 to be a single structure with two parts, one for a floating point thread (204 and 208) and another for an integer thread (206 and 210). Even if this interpretation were not valid, the court has held that integrating two structures into a single structure with no change in functionality is merely engineering choice and not patentable. *In re Larson* 144 USPQ 347 (CCPA 1965).

62. Applicant argues that Yeager does not anticipate claim 22 because the graduation mask pointed out by the Examiner is not a pointer. The included dictionary definition of "pointer" (third definition) defines a pointer to be a physical or symbolic identifier of a unique target. With this definition, the graduation mask of Yeager, which is shown in column 13, lines 12-13 to identify which instructions graduate or retire, is in fact a retire pointer. The graduation mask identifies a unique target, the instructions to graduate or retire.

63. Applicant then argues that a retire row pointer is not disclosed by Yeager. As just described, there is in fact a retire pointer (graduation mask) in the system and since

the instructions to be retired exist in rows of the structure (figure 8), the pointer is a retire row pointer.

64. Applicant argues that the Yeager patent does not anticipate claim 28 because it does not disclose a single structure with two parts, each part to track a different thread but instead discloses two structures each used to track a single thread. As indicated in the arguments above, the Examiner is validly viewing elements 204, 206, 208, and 210 to be a single structure with two parts, one for a floating point thread (204 and 208) and another for an integer thread (206 and 210). Even if this interpretation were not valid, the court has held that integrating two structures into a single structure with no change in functionality is merely engineering choice and not patentable. *In re Larson* 144 USPQ 347 (CCPA 1965).

65. Applicant argues with respect to claim 29 again that the Yeager patent moves a write pointer in increments determined by the number of instructions which graduate during each clock cycle and moves the read pointer in increments by the number of free registers assigned during each clock cycle and thus does not anticipate claim 29. The Examiner recognizes what the Applicant asserts to be true, however this does not mean that the Yeager reference does not anticipate claim 29 as will be described further below. Applicant also argues that Yeager patent does not disclose two pointers set apart by a fixed distance that move in unison up and down a structure to track register allocation of a first thread of a microprocessor, but instead discloses two pointers that move independently. Column 13, lines 1-14 show a set of pointers (write pointer, read pointer, and graduation mask or retire pointer). Column 14, lines 54-55 show that the

write pointer is incremented by the number of instructions that graduate (retire) each cycle. Column 15, lines 1-2 show that the read pointer is incremented by the number of free registers assigned each cycle. In the case of a cycle where the same number of instructions retire as the number of free registers that are assigned, the read and write pointers move in unison up and down said structure (up when incrementing and down when both pointers wrap around the circular FIFO (column 12, lines 58-60)). Further, at any given point there is a fixed distance between the read and write pointers. While there are times where the read and write pointers do not move in unison, there clearly are times when the pointers do move in unison. Applicant has not claimed where the pointers move in unison at all times.

66. Applicant argues that the Yeager patent does not anticipate claim 49 because it does not disclose a single structure with two parts, each part to track a different thread but instead discloses two structures each used to track a single thread. As indicated in the arguments above, the Examiner is validly viewing elements 204, 206, 208, and 210 to be a single structure with two parts, one for a floating point thread (204 and 208) and another for an integer thread (206 and 210). Even if this interpretation were not valid, the court has held that integrating two structures into a single structure with no change in functionality is merely engineering choice and not patentable. *In re Larson* 144 USPQ 347 (CCPA 1965).

Conclusion

67. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or

patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

68. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The references cited in the previous Office Action remain pertinent and are cited herein by reference.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane F Gerstl whose telephone number is (571) 272-4166. The examiner can normally be reached on M-F 6:45-4:15 (First Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane F Gerstl
Examiner
Art Unit 2183

SFG
November 8, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100